

Guided Statistical Workflows with Interactive Explanations and Assumption Checking

Yuqi Zhang*
New York University

Adam Perer†
Carnegie Mellon University

Will Epperson‡
Carnegie Mellon University

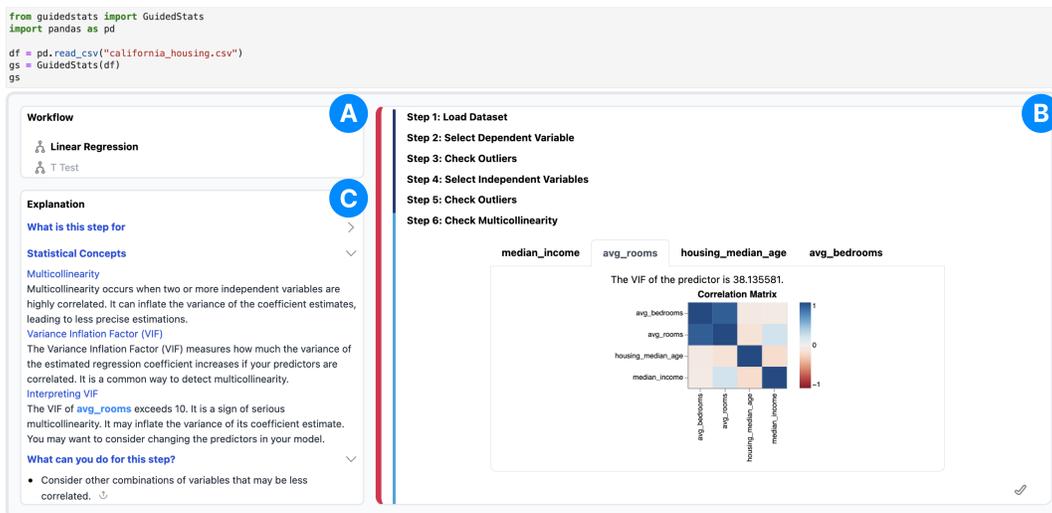


Figure 1: *GuidedStats* helps users perform statistical analyses with guided workflows. (A) A user selects to run the linear regression workflow. (B) *GuidedStats* guides the user through the nine steps of this workflow. Each step either solicits user input or helps users verify assumptions with visualizations and tests of the data. (C) *GuidedStats* provides explanations for the current step in the workflow. The user is currently on an *assumption checking step*; the explanation shows more information about the statistical concepts in the current step.

ABSTRACT

Statistical practices such as building regression models or running hypothesis tests rely on following rigorous procedures of steps and verifying assumptions on data to produce valid results. However, common statistical tools do not verify users’ decision choices and provide low-level statistical functions without instructions on the whole analysis practice. Users can easily misuse analysis methods, potentially decreasing the validity of results. To address this problem, we introduce *GuidedStats*, an interactive interface within computational notebooks that encapsulates guidance, models, visualization, and exportable results into interactive workflows. It breaks down typical analysis processes, such as linear regression and two-sample T-tests, into interactive steps supplemented with automatic visualizations and explanations for step-wise evaluation. Users can iterate on input choices to refine their models, while recommended actions and exports allow the user to continue their analysis in code. Case studies show how *GuidedStats* offers valuable instructions for conducting fluid statistical analyses while finding possible assumption violations in the underlying data, supporting flexible and accurate statistical analyses.

Index Terms: Data science tools, computational notebooks, analytical guidance

*e-mail: yz9381@nyu.edu

†e-mail: adamperer@cmu.edu

‡e-mail: willepp@cmu.edu

1 INTRODUCTION

Statistical analysis is widely used to guide decision making and make sense of experimental results. However, producing *valid* statistical estimates requires following particular statistical procedures with specific data assumptions [2, 23]. In practice, even basic analysis methods like null hypothesis testing or linear regression models are misused, leading to potentially incorrect analysis results [7, 12, 24, 25].

One of the ways statistical methods are misused is when assumptions are not properly verified [24]. All statistical models rely on assumptions about the input variables. If these assumptions are heavily violated, the statistical estimates may be biased and inconsistent, potentially leading to incorrect inferences and misguided decision making [7, 18]. For example, during a two-sample T-test, analysts may neglect to check if the variances of the two samples are equal. The t statistic is calculated differently depending on whether variances are equal or not, and an incorrect t statistic might lead to a false rejection of the null hypothesis.

Correctly using statistical methods is difficult in part because the resources for *how* to do the analysis are separate from *where* the analysis takes place. Numerous textbooks and frameworks exist describing the assumptions that must be met when using statistical tools like hypothesis tests or linear regressions [9, 13]. However, many analysts perform their analyses in coding environments like computational notebooks, where they can manipulate, visualize, and run statistics on their data using various packages. These code-based analysis environments are preferred over GUI-based tools since programming affords more flexibility in the analysis [20, 22]. Despite their flexibility, these statistical analysis packages can be easily misused as they offer little assistance in ensuring correct ma-

nipulations. Providing guidance can bridge the knowledge gap necessary for users to make analysis progress and thus perform correct statistical practices [4, 5].

In this paper, we introduce *GuidedStats*, an interactive system within computational notebooks to guide analysts through statistical workflows. *GuidedStats* contains different workflows for common statistical procedures that walk a user through steps to specify inputs for their analysis and check necessary assumptions. *GuidedStats* offers guidance in two primary ways: by walking a user through the proper steps in a workflow and explaining how to interpret each step. Step explanations include the step’s objective, statistical concepts, and how to interpret the results. After checking assumptions, *GuidedStats* offers suggestions based on the results. Each suggestion includes actions that can populate later inputs or export code to transform the data to meet the assumptions.

As a Jupyter Notebook Widget, *GuidedStats* allows fluid exchanges between the user interface and the coding environment. *GuidedStats* takes in a dataset for analysis; once the statistical workflow is complete, users can export the results back to code. During the analysis, users can explore or change the dataset used by *GuidedStats* with code.

We present two use cases where *GuidedStats* helps users find assumption violations and possible actions for the assumption checking results on linear regression and two sample T-test workflows. Our code is open source and available for use¹.

In summary, our paper makes the following contributions.

1. A framework for guided statistical analysis with automatic assumption checking and explanations. Assumption checks help users ensure their analysis meets expected assumptions, while explanations help users understand the concepts and interpret the results at each step. Assumption violations can be corrected through recommended actions on the data.
2. *GuidedStats*, a computational notebook extension that incorporates these guidance elements into an interactive system for guided statistical workflows and supports hand-off between the interface and code.

2 RELATED WORK

Commonly used statistical analysis tools such as SPSS, SAS, Stata, and R, provide UIs for statistical analysis, but they often lack sufficient explanations and documentation for users to effectively perform and understand statistical procedures [11]. This lack of support limits analysts in understanding the underlying concepts behind the methods they use [27]. Furthermore, these tools may not verify the assumptions of the statistical models the analysts choose, which leads to unreliable statistical estimates.

Researchers have developed statistical tools that offer step-by-step guidance and verify assumptions during analysis. *StatHand* is a web-based application designed to help students identify appropriate statistical tests and provide guidance on using statistical tools and explaining statistical concepts [3]. However, *StatHand* can not perform statistical analysis within its interface or offer guidance on decisions made during the analysis. *Statsplorer* allows users to perform statistical tests within its web-based application, explaining statistical concepts and verifying necessary test assumptions [26]. While these tools can explain statistical concepts and interpret model results, they do not guide decision-making during analyses, such as managing assumption checking results. In contrast, *GuidedStats* provides guidance specific to the current dataset and recommends actionable steps for handling assumption violations.

Tea and *Tisane* offer high-level declarative languages for authoring statistical analyses [14, 15]. They help users specify hypotheses, study designs, and variable relationships, automatically verifying assumptions and inferring valid models. While *Tisane* provides

suggestions for assumption violations, it focuses on explaining concepts in assumptions, requiring manual implementation and risking misuse. In contrast, *GuidedStats* guides the entire statistical analysis process by explaining each step’s objectives and suggesting actions based on assumption verification results. *GuidedStats* also supports iterative model development, allowing users to revise decisions, tightening the feedback loop of model refinement.

Current interactive statistical tools exist apart from where many analyses actually take place: computational notebooks. Computational notebooks integrate text, code, and visualization into a single document, supporting flexible and iterative analyses [20, 22]. Prior systems extend computational notebooks to better support revisiting history in analysis and data profiling [6, 17, 28]. Likewise, *GuidedStats* is a computational notebook extension to support easily jumping between GUI and code. A user provides an initial dataset to *GuidedStats*, which can be later updated if the data needs transformation after an assumption violation is discovered. At the end of an analysis, the results can be exported back to code.

3 *GuidedStats*: STEP-BY-STEP STATISTICAL WORKFLOWS

GuidedStats is a computational notebook extension that has statistical **workflows** comprised of **steps** and **explanations**.

3.1 Statistical workflows

To start using *GuidedStats*, a user selects which statistical **workflow** they want to use. Each workflow represents a statistical analysis and guides users through the steps in the workflow. *GuidedStats* currently contains two workflows: Linear Regression and T-Tests, but can be extended in the future to other statistical workflows. Workflow steps are tasks such as choosing input variables and checking model assumptions. Other statistical methods can be formulated as workflows in *GuidedStats* by specifying the necessary steps to gather inputs and assumptions.

Workflows also support the iterative refinement of decision choices. Every time a previous step is edited, the following steps are rerun with the new parameters, enabling quick iteration when testing different decision choices. Additionally, a workflow manages imports and exports for the analysis. As input, the workflow ingests the initial dataset. Once the analysis is complete, a user can export the created model and the report of the results. During the analysis, workflows can export code for each ongoing step and the dataset. If a dataset is transformed in the notebook, it can be re-imported back to the workflow, which will re-run the analysis up to the current step.

3.2 Steps in a workflow

In *GuidedStats*, workflows are comprised of individual **steps**. Each step represents a sub-task in the analysis. For instance, as shown in Figure 1, the linear regression workflow includes steps for identifying independent variables (Step 4) and verifying the assumption of outliers (Step 3 and 6). The steps, assumptions, and visualizations in the current *GuidedStats* workflows are adapted from statistics textbooks [2, 10, 23] and further refined using examples from online analysis packages like scikit-learn [19] or Vega-Lite [1].

There are three kinds of workflow steps in *GuidedStats*: User Input Steps, Result Presentation Steps, and Assumption Checking Steps (Table 1). User Input Steps enable users to set parameters in the analysis, such as selecting independent variables. Result Presentation Steps are always the last step in an analysis and present the results of a workflow. Assumption Checking Steps verify the assumptions of a workflow. For example, the linear regression workflow includes checks for outliers and multicollinearity.

Assumption Checking Steps are important for ensuring valid statistical models and informing user actions. For instance, if the data fails the normality assumption in a two-sample T-test, users might transform the data or choose a non-parametric test instead. Assumption Checking Steps remind users of important assumptions,

¹<https://github.com/cmudig/GuidedStats>

Step Type	Purpose	Explanation Components
User Input Step	Require user inputs from the interface to specify their decisions	<ol style="list-style-type: none"> 1. <i>Objective</i>: purpose of the step 2. <i>Statistical Concepts & Interpretation</i>: explanations of important statistical concepts used in the step
Result Presentation Step	Presents the outcomes of the statistical analysis	<ol style="list-style-type: none"> 1. <i>Objective</i>: purpose of the step 2. <i>Statistical Concepts & Interpretation</i>: interpretation of the modeling results
Assumption Checking Step	Verify the assumptions of models to ensure the validity of statistical estimates	<ol style="list-style-type: none"> 1. <i>Objective</i>: purpose of the step 2. <i>Statistical Concepts & Interpretation</i>: interpretation of the results of tests used to verify the assumptions 3. <i>Suggested Actions</i>: possible actions if the assumption is violated

Table 1: The types, purpose, and explanation components of steps. See § 3.3 for the example of each.

which they can choose to ignore depending on the analysis context. For example, with a large sample size, the robustness of the T-test allows users to overlook non-normal distributions of two groups. Violations in Assumption Checking Steps are accompanied by Suggested Actions that users can take to potentially fix the violation.

3.3 Step Explanations

Explanations are offered for each step in *GuidedStats* and have three potential components: the objective of the step, the statistical concepts & interpretation, and suggested actions. Table 1 details the components of the explanation for each type of step.

All steps have an objective explanation and a statistical concept explanation. The objective of the step clarifies why the step is included in the workflow and its importance to the overall statistical analysis. For instance, the step for stating the hypothesis in the two sample T-test has the objective “Formulate the null and alternative hypotheses for the T-test and define the Type I Error (alpha)”.

The statistical concepts and interpretation help users interpret the results in each step. This type of explanation is included in each step but is particularly important for the result presentation step. For example, in the result presentation for a linear regression, the explanation shows how to interpret the effect of the coefficients and the metrics for measuring the fitness of model, such as R^2 .

Suggested actions are provided only for Assumption Checking Steps. Like the steps themselves, these actions are derived from standard statistical textbooks [2, 10, 23]. Actions take two forms: setting up the inputs for parameters in future steps or generating code to fix an assumption violation inspired by prior approaches for exportable analysis snippets [6]. For example, when checking the linear regression assumption of multicollinearity, a suggestion is presented as “consider other combinations of variables that are less correlated”. Clicking on that action will produce a code cell to drop the variables with the highest correlation.

4 USE CASES

In this section, we describe two example statistical analyses using *GuidedStats* with real-world datasets: a multivariate linear regression and a two-sample T-test. In these examples, the analysts are novices in statistics but aim to perform reliable statistical analyses for their reports. These analyses illustrate how they can benefit from the design and guidance of *GuidedStats* during the analyses.

4.1 Linear Regression

In our first use case, we discuss an analyst who wants to understand how housing characteristics such as housing age and neighborhood income affect house value. For their analysis, they use the California Housing Prices Dataset, which contains 20,640 observations of the median house value for different blocks of houses in California [16]. They want to use a linear regression model to help them estimate the impact of different features on the median house value, so turn to *GuidedStats* to create the model.

First, the analyst imports the dataset and selects the Linear Regression Workflow. The workflow contains 9 steps, including 3 assumption checking steps to check the core assumptions of linear

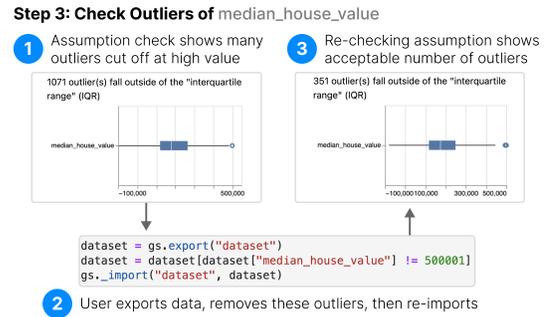


Figure 2: *GuidedStats* supports an interactive loop of assumption checking, editing the data, then re-verifying assumptions.

regression: no large outliers in the independent variables or dependent variables, and no perfect multicollinearity [23]. After loading the data, the second step is variable selection, where the analyst selects `median_house_value` as the dependent variable from the displayed list of columns. *GuidedStats* then moves to Step 3 to help the analyst verify the first assumption: that no large outliers exist in the target variable. This step plots the distribution of `median_house_value` and indicates that 1,071 outlier points lie outside the interquartile range (a common method for checking outliers). The explanation block of this step suggests actions to deal with these outliers, such as removing the outliers or applying data transformation to the column, like a log transform. The analyst follows the suggestion to investigate the column first and clicks the action button of this suggestion. *GuidedStats* exports code for descriptive analysis on `median_house_value` to a new cell below. The output reveals that the most frequent value is 500,001, where `median_house_value` was manually clipped. The analyst then removes rows with `median_house_value = 500,001`, and re-imports the transformed dataset to *GuidedStats* (Figure 2).

Next, in Step 4 (Variable Selection Step), the analyst selects `median_income`, `total_rooms`, `total_bedrooms`, `housing_median_age` as independent variables to see how they impact the median housing price. In the next step, *GuidedStats* once again helps the analyst check the assumption that no large outliers exist in the independent variables either. Inspecting the output from this step shows that some selected variables, `total_rooms` and `total_bedrooms`, have a large amount of outliers and thus might jeopardize the validity of the regression model if used. After reviewing the dataset description, the analyst decides to create more meaningful variables `avg_rooms` and `avg_bedrooms` as the average of the rooms and bedrooms for all houses in the block and use these as independent variables.

After ensuring no large outliers, *GuidedStats* moves to another assumption checking step to check for multicollinearity by calculating Variance Inflation Factor (VIF) of each independent variable. This metric measures the correlation among the independent variables and its effect on accuracy. *GuidedStats* shows that `avg_rooms` and `avg_bedrooms` have high VIFs (38.14 and 30.57), indicating

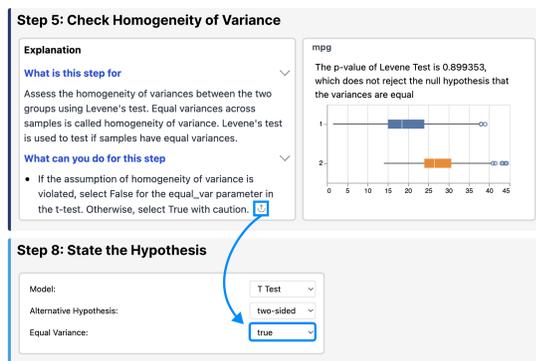


Figure 3: For assumption checking steps, like the homogeneity of variance in the T-test workflow, *GuidedStats* recommends potential actions based on the results of assumption checks. In this example, the check suggests the variances are equal and the user can select the **action** to pre-set this parameter to True in the later model specification step.

significant multicollinearity, though they may still be valid depending on the context (Figure 1).

The analyst notes this issue and proceeds to the next two user input steps: data splitting and model hyperparameter selection. In the model specification step, they choose a simple linear regression. In the final model evaluation step, *GuidedStats* displays the coefficients and goodness-of-fit measures, such as R^2 . The R^2 value, at 49.4%, indicates that approximately half of the variance in `median_house_value` is explained by selected independent variables. The positive coefficient for `avg_bedrooms` suggests that the median house price increases as the average number of bedrooms increases. Conversely, a negative coefficient for `avg_rooms` indicates as the average number of rooms increases, the median house price decreases. This somewhat contradictory result is likely due to the multicollinearity observed earlier, which may undermine the model's interpretive power - a major concern. This highlights the need for further refinement of the model.

The analyst continues iterating on different combinations of independent variables in Step 4 (Variable Selection Step) while keeping other decision choices unchanged. They finalize a model with independent variables `median_income`, `avg_rooms`, `housing_median_age` and `households`. The model produced with these features has a slightly lower R^2 (0.47 compared to 0.49) than before. However, the analyst would rather use this newer model since there is no multicollinearity between the features.

4.2 Two Sample T-Test

In our second use case, we describe an analysis of car data to test if miles per gallon (mpg) of cars differs between regions. This analysis uses the cars dataset, which contains 398 observations with 8 attributes about cars [21]. Our analyst tests the null hypothesis that the cars have no difference in mpg between the US and Europe.

The analyst first imports the dataset into *GuidedStats* and selects the two sample T-test workflow. This workflow has 9 steps, with 4 of them to check the 3 core assumptions of a T-test. The workflow begins by asking the analyst for the variable where they select `mpg`. Next, *GuidedStats* guides the analyst to check the first assumption of a T-test: lack of outliers. Here, *GuidedStats* displays the same distribution overview used in the prior use case, showing only 1 outlier in `mpg`. Finding this acceptable, the analyst moves on to the next step, which is to select the two groups to compare in the T-test. They first select the column `origin`, then the values for US and Europe as the groups to compare.

GuidedStats continues to the next step to check the next assumption for T-Tests: the homogeneity of variance between groups. Since the analyst is new to T-Tests, they are unsure what this as-

sumption is checking and turn to the explanation block to learn more. *GuidedStats* explains that homogeneity of variance assesses whether the comparing groups have similar variances and can be evaluated using Levene's test [8] (Figure 3 top). *GuidedStats* plots the distributions of the two groups along with Levene's test result. The p-value of Levene's test is 0.90 and doesn't support rejecting that variances are equal. As suggested, the analyst clicks on its action button to pre-select the parameter for "equal variance" of the T-test as True in the later model specification step (Figure 3 bottom).

In Steps 6 and 7 (Assumption Checking Steps), *GuidedStats* helps the analyst check the final assumption of T-test on whether the two groups are normally distributed. The analyst initially observes that the distributions of the two groups are right-skewed in the density plot and is concerned that this assumption may be violated. However, *GuidedStats* includes a suggestion: "If the sample size is large enough (greater than 30 as a rule of thumb), the T-test is robust to violations of normality". When the analyst clicks on the action button for this suggestion, *GuidedStats* shows them a notice stating that the sizes of two groups are both sufficiently large to be treated as normal. Therefore, the analyst bypasses the assumption and proceeds to Step 8 (Model Step). Here they select "two-sided" as the alternative hypothesis and retain the pre-selected "equal variance" parameter set to True from the earlier action. In the last step for evaluation, *GuidedStats* plots the mean difference of two groups and shows a t-statistic of -8.9147 with p-value close to zero. Consequently, the analyst rejects the null hypothesis and concludes that there is no difference in mpg between cars from the US and Europe.

In these analyses, *GuidedStats* helps the analysts identify and deal with outliers and multicollinearity in the linear regression model and verifies the assumptions of equal variance and normality in the two-sample T-test. The interpretation and suggested actions, such as interpreting high VIF scores and presetting the "equal variance" parameter, would have been difficult and time-consuming to manage without the support of *GuidedStats*.

5 DISCUSSION AND FUTURE WORK

In statistical analysis, verifying assumptions is important to ensure the validity and reliability of results. *GuidedStats* provides a framework incorporating assumption verification with actionable explanations in the analysis workflow. It ensures that the assumptions of statistical models are well-supported by test results, visualizations, and explanations with suggested actions.

However, some assumptions can not be tested numerically as they extend beyond the dataset itself. For example, linear regression assumes variables are independently and identically distributed (i.i.d.) [23]. To verify this, users must know the data collection process prior to their analyses. While *GuidedStats* could suggest examining the randomness of data collection or export exploratory data analysis code, these suggestions are not actionable from the system as they are not derived from the dataset or research design. Adding such steps could increase the ambiguity of statistical analyses and introduce new statistical concepts without clear interpretation or actionable guidance. *GuidedStats* focuses on the "must-have" assumptions for each workflow. Other assumptions that are not numerically testable or "optional" for statistical validity (such as the normality of error terms for statistical derivation) are not incorporated. Future work might explore how to also incorporate these other less-strict versions of assumptions.

As a computational notebook extension, *GuidedStats* facilitates transitioning between code-based analysis and a user interface. Future work could enhance the pairing of code-based and GUI-based analysis such as monitoring users' code and alerting them to potential statistical issues. Additionally, other interfaces such as natural language chat-based tools could allow users to ask questions about their workflow or data, further augmenting statistical programming.

ACKNOWLEDGMENTS

Many thanks to the anonymous reviewers for their insightful suggestions. We also thank Jiayuan Song, Richard Lin, Zhuoqing Li, and Zhuo Zeng for their valuable feedback and inspiration during the development of the system.

REFERENCES

- [1] Example Gallery. Vega-Lite. Accessed: Apr. 20, 2024. [Online]. Available: <https://vega.github.io/vega-lite/examples/>. 2
- [2] A. Agresti and B. Finlay. *Statistical methods for the social sciences*. Pearson Prentice Hall, Upper Saddle River, N.J., 4 ed., 2009. 1, 2, 3
- [3] P. J. Allen, L. D. Roberts, F. D. Baughman, N. J. Loxton, D. Van Rooy, A. J. Rock, and J. Finlay. Introducing StatHand: A cross-platform mobile application to support students' statistical decision making. *Frontiers in Psychology*, 7, 2016. doi: 10.3389/fpsyg.2016.00288 2
- [4] D. Ceneda, C. Collins, M. El-Assady, S. Miksch, C. Tominski, and A. Arleo. A heuristic approach for dual expert/end-user evaluation of guidance in visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):997–1007, 2024. doi: 10.1109/TVCG.2023.3327152 2
- [5] D. Ceneda, T. Gschwandtner, T. May, S. Miksch, H.-J. Schulz, M. Streit, and C. Tominski. Characterizing guidance in visual analytics. *IEEE Transactions on Visualization and Computer Graphics*, 23(1):111–120, 2017. doi: 10.1109/TVCG.2016.2598468 2
- [6] W. Epperson, V. Gorantla, D. Moritz, and A. Perer. Dead or alive: Continuous data profiling for interactive data science. *IEEE Transactions on Visualization and Computer Graphics*, 30(1):197–207, 2023. doi: 10.1109/TVCG.2023.3327367 2, 3
- [7] A. F. Ernst and C. J. Albers. Regression assumptions in clinical psychology research practice—a systematic review of common misconceptions. *PeerJ*, 5, 2017. doi: 10.7717/peerj.3323 1
- [8] J. L. Gastwirth, Y. R. Gel, and W. Miao. The impact of Levene's test of equality of variances on statistical theory and practice. *Statistical Science*, 24(3):343 – 360, 2009. doi: 10.1214/09-STS301 4
- [9] G. J. Hahn and W. Q. Meeker. Assumptions for statistical inference. *The American Statistician*, 47(1):1–11, 1993. doi: 10.1080/00031305.1993.10475924 1
- [10] C. Hanck, M. Arnold, A. Gerber, and M. Schmelzer. *Introduction to Econometrics with R*. Universität Duisburg-Essen, 2021. 2, 3
- [11] C. B. Hodges, B. M. Stone, P. K. Johnson, J. H. Carter, C. K. Sawyers, P. R. Roby, and H. M. Lindsey. Researcher degrees of freedom in statistical software contribute to unreliable results: A comparison of non-parametric analyses conducted in SPSS, SAS, Stata, and R. *Behavior Research Methods*, 55:2813 – 2837, 2023. doi: 10.3758/s13428-022-01932-2 2
- [12] R. Hoekstra, H. Kiers, and A. Johnson. Are assumptions of well-known statistical techniques checked, and why (not)? *Frontiers in Psychology*, 3, 2012. doi: 10.3389/fpsyg.2012.00137 1
- [13] E. Jun, M. Birchfield, N. De Moura, J. Heer, and R. Just. Hypothesis formalization: Empirical findings, software limitations, and design implications. *ACM Trans. Comput.-Hum. Interact.*, 29(1), jan 2022. doi: 10.1145/3476980 1
- [14] E. Jun, M. Daum, J. Roesch, S. Chasins, E. Berger, R. Just, and K. Reinecke. Tea: A high-level language and runtime system for automating statistical analysis. In *Proceedings of the 32nd Annual ACM Symposium on User Interface Software and Technology*, UIST '19, p. 591–603. Association for Computing Machinery, New York, NY, USA, 2019. doi: 10.1145/3332165.3347940 2
- [15] E. Jun, A. Seo, J. Heer, and R. Just. Tisane: Authoring statistical models via formal reasoning from conceptual and data relationships. In *Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems*, CHI '22. Association for Computing Machinery, New York, NY, USA, 2022. doi: 10.1145/3491102.3501888 2
- [16] R. Kelley Pace and R. Barry. Sparse spatial autoregressions. *Statistics & Probability Letters*, 33(3):291–297, 1997. doi: 10.1016/S0167-7152(96)00140-X 3
- [17] M. B. Kery, D. Ren, F. Hohman, D. Moritz, K. Wongsuphasawat, and K. Patel. mage: Fluid moves between code and graphical work in computational notebooks. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, UIST '20, p. 140–151. Association for Computing Machinery, New York, NY, USA, 2020. doi: 10.1145/3379337.3415842 2
- [18] J. W. Osbourne and E. Waters. Four assumptions of multiple regression that researchers should always test. *Practical Assessment, Research and Evaluation*, 8, 2002. doi: 10.7275/r222-hv23 1
- [19] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and Édouard Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12(85):2825–2830, 2011. doi: 10.5555/1953048.2078195 2
- [20] J. M. Perkel. Why jupyter is data scientists' computational notebook of choice, Oct 2018. 1, 2
- [21] R. Quinlan. Auto MPG. UCI Machine Learning Repository, 1993. doi: 10.24432/C5859H 4
- [22] A. Rule, A. Tabard, and J. D. Hollan. Exploration and explanation in computational notebooks. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, p. 1–12. Association for Computing Machinery, New York, NY, USA, 2018. doi: 10.1145/3173574.3173606 1, 2
- [23] J. H. Stock and M. W. Watson. *Introduction to Econometrics (Fourth Edition)*. Pearson, 2019. 1, 2, 3, 4
- [24] M. S. Thiese, Z. C. Arnold, and S. D. Walker. The misuse and abuse of statistics in biomedical research. *Biochemia Medica*, 25(1):5–11, 2015. 1
- [25] J. B. Vornhagen, A. Tyack, and E. D. Mekler. Statistical significance testing at CHI PLAY: Challenges and opportunities for more transparency. In *Proceedings of the Annual Symposium on Computer-Human Interaction in Play*, CHI PLAY '20, p. 4–18. Association for Computing Machinery, New York, NY, USA, 2020. doi: 10.1145/3410404.3414229 1
- [26] C. Wacharamanotham, K. Subramanian, S. T. Völkel, and J. Borchers. Statsplorer: Guiding novices in statistical analysis. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, p. 2693–2702. Association for Computing Machinery, New York, NY, USA, 2015. doi: 10.1145/2702123.2702347 2
- [27] C. Wild and M. Pfannkuch. Statistical thinking in empirical enquiry. *International Statistical Review*, 67(3):223–248, 1999. doi: 10.1111/j.1751-5823.1999.tb00442.x 2
- [28] Y. Wu, J. M. Hellerstein, and A. Satyanarayan. B2: Bridging code and interactive visualization in computational notebooks. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, p. 152–165. Association for Computing Machinery, New York, NY, USA, 2020. doi: 10.1145/3379337.3415851 2